Introduction to Artificial Intelligence

VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

COMP307

Planning and Scheduling:

Tutorial 2

Dispatching Rule

- Dispatching Rule: a rule to select one action in each state
  - Considering ONLY the earliest applicable actions (non-delay)
  - Assigning a priority to each earliest action by a priority function
    - Selecting the action with the highest priority

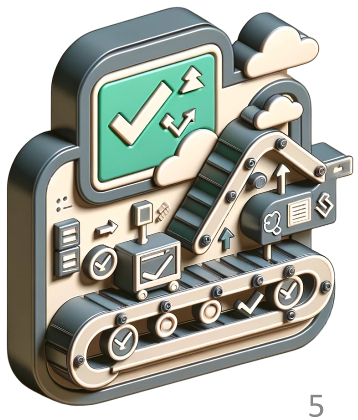
- An example: Shortest Processing Time (SPT)
  - Always select the shortest processing time
  - Priority of Process(o, m, t) is **-ProcTime(o)**

Generate a Schedule by Dispatching Rule

- Step 1: Initialize state
  - empty schedule, all operations unprocessed, time = 0, machine idle time = 0, first operation ready time = **arrival time**, other operation ready time = ∞
- Step 2: Find the earliest applicable actions;
- Step 3: Select the next action by the dispatching rule
- Step 4: Add the selected action into the schedule, update the state
- Step 5: If all operations are processed, stop. Otherwise, go to step 2.

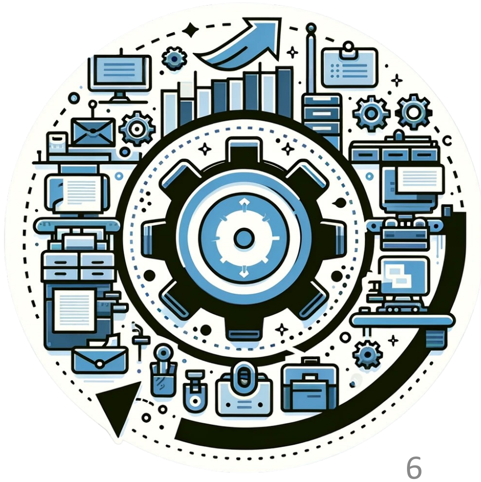
Advantage of Dispatching Rule

- Can be apply at ANY time point to change the remaining schedule
  - Initial state = current state
  - But only need at critical time point (a machine becomes idle, an operation becomes ready)
- Select ONLY the next action to be taken, NO need to generate the entire remaining schedule
- Very quick in real time, can handle dynamic environment very well
  - At each time point, complexity = #unprocessed ops * O(priority)



Generate Schedule by SPT

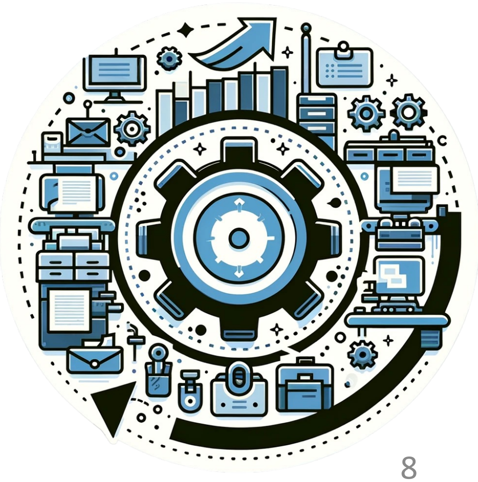
	Arrive	ProcTime		
		AddEngine	AddWheels	Inspect
Job 1	0	30	30	10
Job 2	0	60	15	10
Job 3	10	20	30	10



Handwritten solution

Generate Schedule by FCFS

	Arrive	ProcTime		
		AddEngine	AddWheels	Inspect
Job 1	0	30	30	10
Job 2	0	60	15	10
Job 3	10	20	30	10



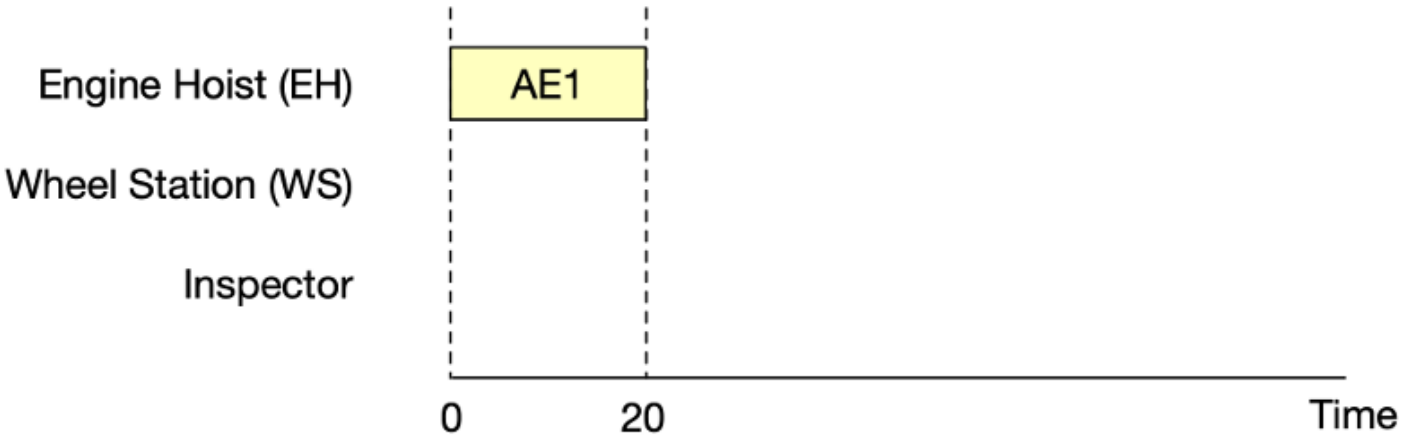
Handwritten solution

Question

- For the car manufacturing scheduling problem, consider 3 jobs as summarized below:

	Arrival Time	Processing Time		
		AddEngine (AE)	AddWheels (AW)	Inspect (Ins)
Job 1	0	20	30	15
Job 2	0	45	20	20
Job 3	10	25	30	10

- Given the partial schedule below:



- Select the next action by the Shortest Processing Time (SPT) rule. Show your working.

Vehicle Routing

- Problem:
  - A graph with the node set and the edge set
  - A special depot node
  - Each edge has a cost (length, travel time, …)
  - Each node except depot has a demand (customer demand)
  - Vehicles with limited capacity
- Find a **solution**:
  - A set of routes, each for a vehicle
  - Each node is visited exactly once
  - Each route starts and ends at the depot (cycle)
  - The total demand of the nodes in each route does not exceed capacity
- **Objective**: minimize the total cost of the routes

Vehicle Routing is Hard

- Too many solutions
  - 10 nodes (excluding depot), 1 vehicle (TSP)
  - 3.6 million solutions
- **NP-hard**
  - Cannot guarantee to find the optimal solution in reasonable time

- How to Solve Vehicle Routing
  - Heuristics: Search for a reasonably good solution in a given (short) time
  - Introduced two heuristics in the lectures:
    - Nearest Neighbor heuristic
    - Savings heuristic

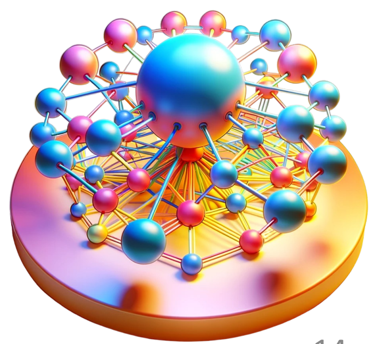
Question

- For a vehicle routing problem with one depot and 5 customers (customer locations). Assume
  - Each customer has a demand of 1.
  - The maximum capacity of each car is 4.

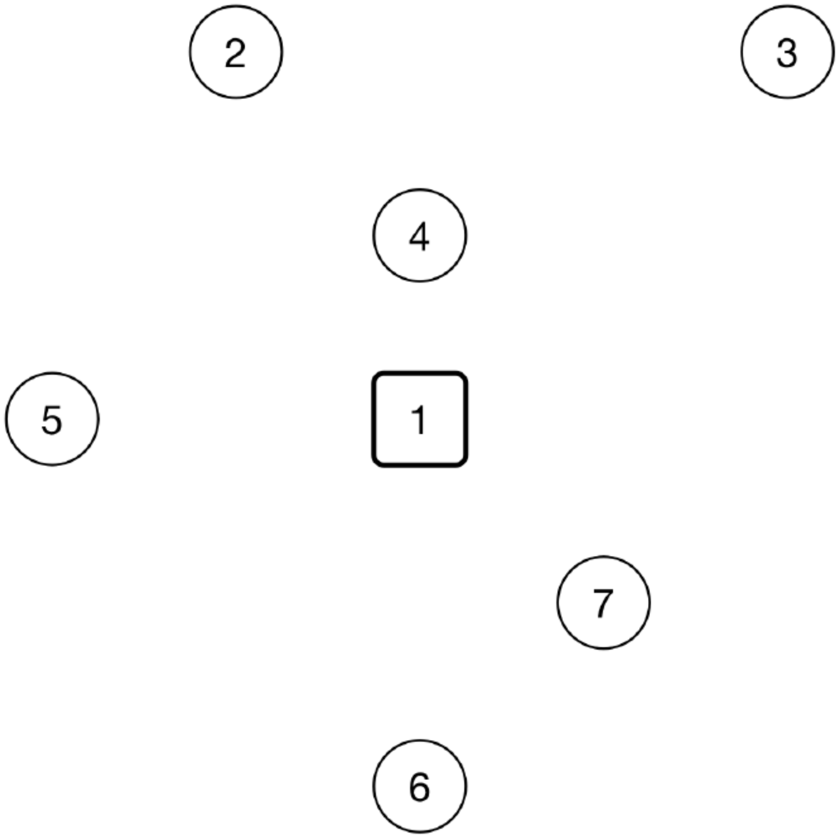
- What is the total number of possible solutions to this vehicle routing problem?

Nearest Neighbor Heuristic

1. Initialize a solution: a route starting from the depot
2. Append the nearest feasible node to the end of the current route
   - Unvisited
   - After inserting the node, the total demand of the route does not exceed the capacity
3. If no feasible node is found for the current route, then close the current route (return to the depot) and create a new route starting from the depot
4. Repeat 2. and 3. until all nodes are visited



Generate VRP Solution

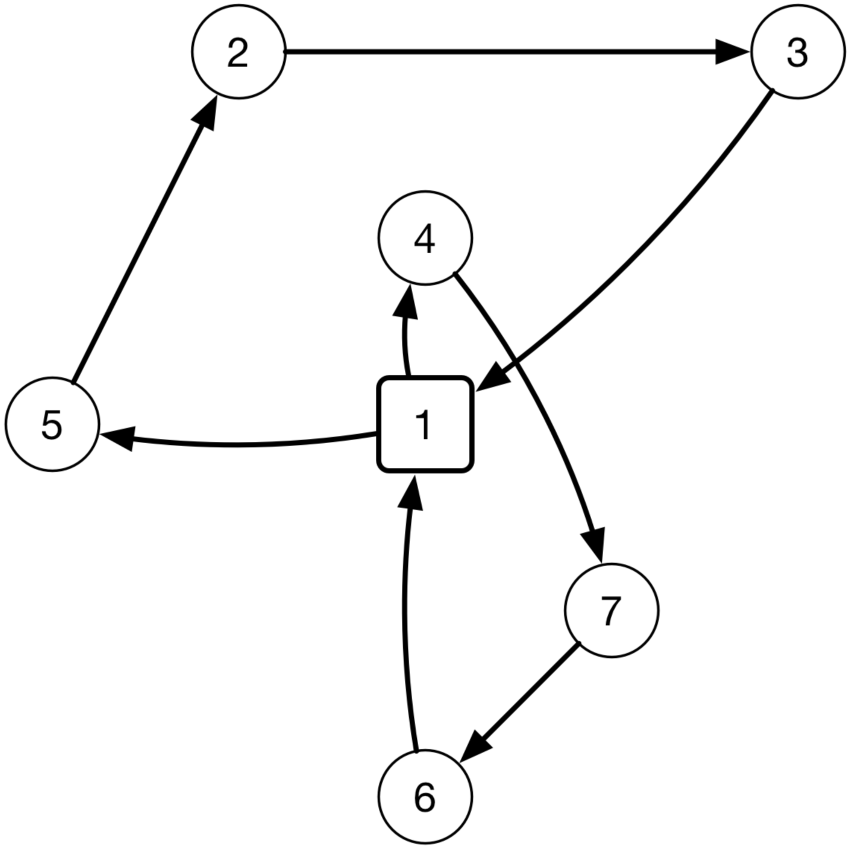


	1	2	3	4	5	6	7
1	0	2.2	2.8	1	2	2	1.4
2	2.2	0	3	1.4	2.2	4.1	3.6
3	2.8	3	0	2.2	4.5	4.5	3.2
4	1	1.4	2.2	0	2.2	3	2.2
5	2	2.2	4.5	2.2	0	2.8	3.2
6	2	4.1	4.5	3	2.8	0	1.4
7	1.4	3.6	3.2	2.2	3.2	1.4	0

Capacity = 3

Handwritten solution

Savings Heuristic



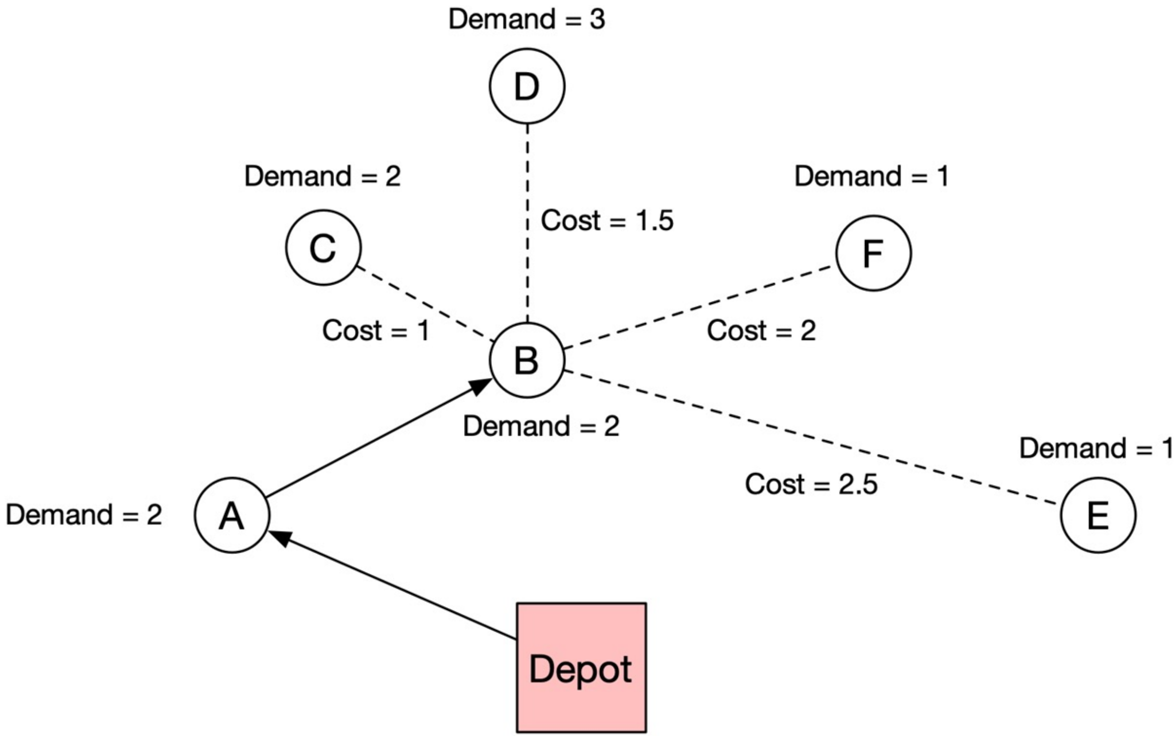
Feasible merges

Merge	Sav
~~(2,3)~~	~~2~~
~~(2,5)~~	~~2~~
~~(6,7)~~	~~2~~
~~(2,4)~~	~~1.8~~
~~(3,4)~~	~~1.6~~
…	…
~~(4,7)~~	~~0.2~~
…	…

Merge	Sav
(2,3)	2
(2,5)	2
(6,7)	2
(2,4)	1.8
(3,4)	1.6
…	…
(4,7)	0.2
…	…

Question

- Consider the vehicle routing with the partial solution [Depot, A, B]. Assume
  – Vehicle capacity is 5
  – Remaining nodes to visit is C, D, E, and F
  – Use the nearest neighbor heuristic



- List the next two steps (nodes to visit) selected by the nearest neighbor heuristic.